

Open-Apple

October 1988
Vol. 4, No. 9

ISSN 0885-4017

newsstand price: \$2.50

photocopy charge per page: \$0.15

Releasing the power to everyone.

Amazing Apple stories

As happens all too often anymore, I once again find myself trapped in the infamous AppleFest time warp. I'm writing this letter to you before the new product introductions of AppleFest, but you're reading it afterward. I could ask you to pretend AppleFest hasn't happened yet, or I could report that Apple's *supposed* to have a new IIc (the IIc Plus) and the real ProDOS 16 (GS/OS) up its sleeves and hope it really does. Or I could just ignore the hubbub in San Francisco and tell you some amazing Apple stories.

For example, have you heard that Apple II software developers all received the "Apple II System Disk, Version 3.1, April 14, 1988" the first week of September? Most of them threw it in the round file, thinking Apple had really screwed up this time, since two months previously these same developers had received the "Apple IIgs System Disk, Version 3.2, June 24, 1988" from Apple. Moreover, they'll be getting Version 4.0 any day now.

But it wasn't a total fiasco—it turns out many developers missed the subtle difference that the disk they got in July was a "IIgs" disk, while the September disk was a "II" disk. Among other goodies, the September disk has the heretofore missing version 1.5 of ProDOS 8 (the July disk has version 1.6) and a new program selector called LAUNCHER.SYSTEM that has a familiar interface but that allows you to select only among System Utilities, FastCopy, Applesoft, or the ancient ProDOS throw-your-hands-up-and-quit routine. Otherwise, everything on it duplicates material on the July IIgs disk, which is now available for downloading on the major online services. Your dealer may even have a copy you can copy. If you don't have it yet, however, wait for 4.0.

Bob Sander-Cederlof has ceased publication of his newsletter, **Apple Assembly Line**, and gone to work for Applied Engineering. He had previously done a number of projects for AE on a consulting basis. This is the second Apple II publication to pass away this year—the other was *Apple User*, a British publication.

Sander-Cederlof's last article was an analysis of Apple's new Basic.system, Version 1.2. It vividly demonstrates why his publication will be missed:

New Version 1.2 of BASIC.SYSTEM, by Bob Sander-Cederlof

(Copyright 1988 by S-C Software, reprinted with permission)

When I receive a new version of something from Apple, my first impulse is to try to find out exactly what they changed. Especially, when for the first time in four years, they update a program so important as Basic.system. And especially when there have been excellent articles published in the last four years clearly describing definite bugs, patches, and work-arounds.

I was very disappointed this morning after carefully analyzing the new version 1.2 of Basic.system. I started by BLOADing the old version 1.1 and they copying it into bank 2 of my IIgs. Then I BLOADed the new version 1.2 and used the Monitor's V-command to compare the two. There were a total of 24 bytes changed. Thirteen were inside the parameter block for a Get_File_Info call, so their value is irrelevant. One is a byte that is never referenced in any way. Three bytes were changed in the title screen, so that you see "1.2" instead of "1.1", and "COPYRIGHT APPLE 1983-87" instead of COPYRIGHT APPLE, 1983-84". That leaves only seven bytes in the total update

whose change has any significance. They have not fixed even ONE of the many published problems in Basic.system!

So what did they fix? The description sheet that came with the update said they were trying to fix a bug in the CATALOG command. A variable they call TOTENT, which happens to be at \$BCB9-BCBA, is used for a counter to control the loop that displays file names and info. When the directory is first opened, the total number of files in the directory is copied into TOTENT. The original intention of the programmer was to decrement TOTENT after reading each file entry in the directory. When the counter reaches zero, the catalog should be finished. Unfortunately, the program did not decrement the counter properly.

To make matters worse, the new code in version 1.2 does not fix the bug. Instead, the patch just omits testing TOTENT altogether. Now if you have a long directory, delete most the files leaving just a few file names in the first few entries, and CATALOG it in Basic.system, it will read all the entries anyway. No real problem, the disk just spins a fraction of a second longer.

The original bug was not a very serious problem either. It only failed when the total number of active files in a directory was a multiple of 256, which seldom happens. In fact, it seldom happens that there are that many files in any one directory, because so many of the utilities and even AppleWorks get confused with large directories. The symptom you would see if you had exactly 256 files in a directory, as I understand it, is that you would get an "OUT OF DATA" error message at the end of the catalog instead of the "number of blocks" line. I suppose that could be unnerving, so the bug should be removed if possible.

The faulty decrement code is at the end of the Read Next Catalog Entry subroutine, at \$B215, and looks like this:

```
B215: DEC $BCB9
B218: BNE $B21D
B21A: DEC $BCBA
```



"OH SURE, \$1.8 MILLION DOLLARS SEEMS LIKE A LOT RIGHT NOW, BUT WHAT ABOUT RANDY? WHAT ABOUT HIS FUTURE? THINK WHAT A COMPUTER LIKE THIS WILL DO FOR HIS SAT. SCORE SOMEDAY."

B21D: RTS

If the initial number in BCB9 (low byte) and BCBA (high byte) is not a multiple of 256, this code will always result in BCBA going negative when the total value has been counted down. But if the initial value IS a multiple of 256, it will take an extra 256 times to count it down to a negative value in BCBA. The end-of-loop test code is at \$B09E:

```
B09E: LDA $BCBA
B0A1: BPL $B078
```

The correct way to decrement the 16-bit value is like this:

```
LDA $BCB9
BNE .1
DEC $BCBA
.1 DEC $BCB9
```

This results in both bytes being zero when it is counted all the way down. Code to test the TOTENT variable for zero already exists at the top of the loop in Basic.system.

```
B070: LDA $BCB9
B073: ORA $BCBA
B076: BEQ $B0A3
```

A little restructuring of the code would result in even fewer bytes being required to do the decrement and loop control correctly. Instead, we have this strange wipe-out instead. Apple went further, and changed the branch at \$B076 to two NOPs, and the error branch following the call to Read Next Catalog Entry to terminate the catalog without error. Very interesting. I wonder if they know something I don't? Maybe the value in the directory that we get TOTENT from is sometimes incorrect? Maybe it is sometimes 0000 when there are really files? Why else NOP-out the instruction at \$B076? Well, I have never yet noticed such a problem. Have you? Notice that, with these patches, if you get a disk error when reading a directory block, CATALOG will terminate without reporting the error: you just will not see the rest of the files.

The description also claimed to fix a problem that caused CATALOG to prematurely terminate if a <space> was pressed after a control-S. I have never noticed any such problem in the old version, and was unable to make it happen today. But sure enough, it doesn't fail that way in the new version either. After all, they didn't change any of that code anyway!

Why didn't Apple confer with Ken Kashmarek, Cecil Fretwell, Sandy Mossberg, Don Worth, Pieter Lechner, Dennis Doms, or others who have been so carefully analyzing ProDOS and Basic.system over the last four years?

Anyway, after all the above is said, maybe you still wish you had version 1.2. If so, you can turn version 1.1 into 1.2 like this:

```
BLOAD BASIC.SYSTEM,TSYS,A$2000
CALL-151
2282:B2      (was B1      )
229A:A0      (was AC      )
22A2:B7      (was B4      )
3A76:EA EA    (was F0 28   )
3A7C:26      (was 3A      )
3A9E:EA A9 FF D0 (was AD BA BC 10)
3D0G
BSAVE BASIC.SYSTEM,TSYS,A$2000
```

Shem the Penman's Guide to Creative Writing, Reasoning, and Programming is one of those rare educational software packages that does things in the classroom with a computer that can't be done any other way. It's the foundation for a semester-long hands-on course designed to improve the writing, reasoning, problem-solving, planning and organizing, group cooperation, and learning skills of students. They also pick up some Applesoft, but that's just a by-product.

Students taking a course based on Shem's guide spend a semester writing an all-text adventure. The author of Shem's guide, Chet Day, is a high-school English teacher, novelist, adventure programmer, and inhabitant of the steamy subculture of computer bulletin boards. His guide is based on an elective course he designed and has given each spring for the last four years.

Day says the course turns teenagers who boast they haven't finished a book since 6th grade into avid readers who spend hours turning the pages of interactive fiction. Among the course's benefits are development of a student's ability to read for details and to think logically.

Shem's guide outlines the course week-by-week. The first week is spent getting familiar with interactive fiction. By the beginning of the second week, each student must have written a story that will form the basis for his or her adventure.

During week two, students design the "map" of their adventure. During weeks three through five, the students write all the text that will make up their games. Then the students spend ten weeks fleshing all this onto an Applesoft skeleton program that Day provides. The final weeks of the course are spent debugging, beta-testing, and creating marketing materials.

Like all the best educational software, *Shem the Penman's Guide* comes with a student manual on disk, where it can be shortened, lengthened, or otherwise modified to fit the needs of individual teachers and schools. The manual includes a thorough, annotated discussion, with examples, of how to write high-quality interactive stories, as well as a complete learn-by-doing tutorial on programming that beginners are able to understand.

The disk also includes a teacher's manual, the programming skeleton, and a sample adventure, "A Day at the Morgue", written especially for 13- to 18-year-old students. The course requires a lab of 80-column-capable Apple IIs (one per student) and AppleWorks. The disk isn't copy-protected, and comes with an unrestricted site license priced at \$65 per school. Demonstration disks are available from all the usual online sources (on GEnie, search for uploads that include the keyword "Shem") or for \$2 from Shem the Penman Software, 625 Smith Dr, Metairie, LA 70005 504-837-0343.

For the past several years I've been convinced that one of the last things the Apple II world needs is another word processor.

My position has been that if you own an Apple II you should own and know how to use AppleWorks. It will do 90 per cent of what you want to do. There are lots of people supporting it with books, newsletters, and training seminars.

If you're one of the people who needs that final 10 per cent, I've said, look at the other word processors that are available and decide if obtaining the final 10 is worth what it will cost you in dollars and in learning time.

In a few cases, of course, it is. My wife, for example, uses both AppleWorks and *APA Manuscript Manager*, a word processor designed specifically for writing professional papers in the format and style required by the American Psychological Association. Many of you use WordWare's *MultiScribe* or Apple's *Apple Writer* in addition to (or instead of) AppleWorks. But in most cases, I've said, the average Apple II user is far better off learning how to do what he or she wants within AppleWorks than chasing the elusive "perfect" word processor.

But now the Bank Street College of Education and Addison-Wesley have brought a new "writing environment" to the Apple II, called *Wordbench*, that has me reconsidering my position. Not to the point that I've actually invested the hours it would take to read the manuals from beginning to end and touch the far limits of the program, mind you, but enough that I haven't yet been able to get the manuals off my desk or the disk out of the IIc I use as a second computer.

Wordbench (note the small "B", thank you), was developed over a three-and-a-half-year period by a team lead by Bank Street's Franklin Smith. It's a lot more than a word processor. Just as AppleWorks integrates a word processor with a database and spreadsheet, *Wordbench* integrates a word processor with a database system oriented around "notes"; outlining software; a spelling checker and thesaurus; a reference tool for creating bibliographies; a print manager that can automatically create a table of contents, sophisticated headers and footers, and footnotes; a "folder" manager for file and disk formatting, renaming, copying, and deletion; and an "add-in" "brainstormer". Additional "add-in" products are possible, using a standard programming interface. *Wordbench* supports macros ("shortcuts") and will use memory expansion cards (recommended).

For screen display, *Wordbench* uses double-high-resolution graphics, which means it displays on your screen exactly what will be printed out on paper, including features such as underlining, boldface, and

superscripts. When printing, however, it uses your printer's standard text rather than graphics-generated fonts. And although it uses the graphics screen and Apple-desktop-like menus, it doesn't use a mouse. All command and block selections are made using cursor keys and Return.

What's so striking about *Wordbench* is how it far departs from today's software trends while still using many of the elements that define those trends. Its use of the graphics screen and its user interface is up-to-date and modern, yet it won't print using fancy fonts or graphics within text. Instead of concentrating on tools for enhancing the look or *form* of the resulting document, the emphasis in *Wordbench* is on tools that enhance the *content* of the document. The form vs. content opposition has always pretty much defined the difference between Macintosh and Apple II computing, but there hasn't been a new Apple II product for months that focuses and delivers on tools for enhancing content as well as *Wordbench* does.

The documentation that comes with the program includes three books, a Tutorial, a User's Guide, and a Reference Manual. The program requires at least 128K in a IIe, IIc, or IIgs (an IBM-compatible version is also available). There is no copy-protection.

The program isn't perfect, of course. It makes your disk drive sound like it's washing clothes when you create a new file and takes almost as long. Scrolling is slow and a built-in keyboard buffer means the screen continues to scroll long after you've released the up or down arrow key.

But the program is new, improvements will come, and it's backed by some very big players in the Apple II community. "A Bank Street spokesman says, '*Wordbench* emerged from Bank Street's interest in developing a product that would address writing as opposed to word processing.' I'm a writer and I'm convinced this is software writers will buy a computer for. The package is \$149 and is distributed to dealers by Ingram Software, Micro D, and SoftKat.

Apple's Developer Services group publishes a miserable little weekly newsletter for developers called *Apple viewpoints*. Most of the feature-length articles published in *Apple viewpoints* imply that if you don't develop software for the Macintosh you're a hairball. IBM itself couldn't design a better piece of propaganda to demoralize Apple II developers. And IBM certainly couldn't aim it at the heart of the Apple II developer community like Apple itself can.

One of the few recent articles that didn't go out of its way to insult Apple II developers was a piece called "The Information-Age Economy" by Apple's president, John Sculley, which appeared in the September 5 issue. Sculley's main point is that as the world moves from an industrial-age economy to an information-age economy, schools have to prepare workers for a different type of job.

"It's not that schools have to train people to do the jobs themselves, what schools really have to do is train people to be trainable," Sculley says. "Institutions used to be measured by their ability to be large, self-sufficient, and stable. Today, the most successful enterprises are measured by their flexibility—their ability to adjust quickly to change..."

Among other things, Sculley seems intent on keeping Apple itself flexible. He's preparing it for the information age by reorganizing it every three months or so. The latest reorganization, announced in mid-August, divides all of Apple into four divisions. They are Apple Products (product marketing, worldwide manufacturing, and research and development), headed by Jean-Louis Gasse; Apple USA (U.S. sales and business marketing, information systems and technology, customer satisfaction, and systems integration), headed by Allan Loren; Apple Europe, headed by Michael Spindler; and Apple Education & Apple Pacific, headed by Del Yocam. The symmetry limps a bit. But it's clearer than ever before that the future of the Apple II (at least for this quarter) is in the hands of Gasse and Yocam.

I don't know if you have any need for any BIG.DUMMYs around your place, but I recently needed one here. I had installed one of these newfangled StatDisks in my wife's enhanced IIe. When she ran AppleWorks, AppleWorks, recognizing the card as a RAMdisk, expanded itself into *all* the remaining memory on the card.

I figured she'd need a little free space on the StatDisk even while running AppleWorks (enough for the QuickSpell custom dictionary to expand into, if nothing else). **Open-Apple** published an AppleWorks

patch previously for limiting the amount of RAM used by AppleWorks (August 1987, page 3.56), but unfortunately, that patch doesn't work with Beagle Bros' *Timeout*. Another possibility was to move a little jumper on the StatDisk itself so that AppleWorks wouldn't recognize it as a RAM card (but so that II-Pluses and unenhanced IIes *would* recognize it as a bootable disk drive), but that static memory is expensive and I wanted to use it as much as possible.

My solution was to write a short Applesoft program that saves a dummy file on the StatDisk. Then it runs an *UltraMacros* task file that starts up AppleWorks and deletes the dummy file, thereby opening up some RAMdisk space. Task files are an *UltraMacros* extension that start up AppleWorks and press its keys. My task file, which I called AW.START, presses Return a couple of times to get past the flip-disk and date prompts, then finds the dummy file and deletes it, then changes the Current Disk Drive to the one my wife normally uses for saving data files. In the end, it stops at the main AppleWorks menu, just as AppleWorks normally does.

This trick would be useful with any RAMdisk, so many of you might be interested in looking at these programs. To start up AppleWorks under this system, put the Applesoft program, BIG.DUMMY, in the same subdirectory with AppleWorks and *UltraMacros*, set your prefix to that subdirectory, and run BIG.DUMMY. This starts a chain reaction that ends up within a memory-expanded AppleWorks that hasn't overtaken the entire RAMdisk.

Here's the Applesoft program:

```
10 REM Save this as BIG.DUMMY
20 PRINT CHR$(4);"BSAVE DUMMY,AS800,LS8000"
30 PRINT CHR$(4);"AW.START"
```

Here's the beginning of the Task File that erases the file created by BIG.DUMMY. You can add other macros to it, but the part shown here should come right at the beginning. If you're not thoroughly familiar with *UltraMacros*, proceed like this:

- * Add a new word processor file to the desktop from scratch.
- * Press open-apple-esc to see the Timeout menu.
- * Select "Macro Compiler", then "Display Current Macro Set".
This will fill the word processor file with your current macros.
- * Add the macro shown below at the top of the file. **Do not enter the comments.** You can, however, enter the commands in a vertical column as I've done here, or you can enter them all in one line as is more typical. Delete any other BA-] macro you might have and any other "start".
- * Select "Macro Compiler" again and "Compile a New Set of Macros."
- * Select "Macro Options" and "Create a Task File".
When it asks for a filename, enter "AW.START". There's no need to enter the prefix of your AppleWorks subdirectory—*UltraMacros* already knows where it is.

```
start
<ba-]>:<all>      this must follow right after "start"
<rt>              pass disk-flip prompt
<rt>              pass date prompt
5<rt>              choose Other Activities
4<rt>              choose Delete files
<$ = "DUMMY">     set up macro 0 for upcoming FIND
<find>            Put cursor on DUMMY
<right>           select it
<rt>              delete it
Y                 Yes, I'm sure
1<rt>              choose Change Current Disk
2<rt>              select list's second drive
<esc>             back to main menu

!                 end of macro
```

One final warning: Before trying to start the BIG.DUMMY chain reaction, run AppleWorks and use "Other Activities", "Select Standard Location of Your Data Disk" to point the 'current disk' at the subdirectory you have AppleWorks and *UltraMacros* in. This is where the dummy file gets saved. The macro won't find it unless you set up AppleWorks to default to that subdirectory. As mentioned before, the end of the macro, as written, changes the default to the second device in AppleWorks' disk drive list. You can change this to whatever you like.



Ask (or tell) Uncle DOS

Corrections and Amplifications: The list of third-party updates required for AppleWorks 2.1 that we published last month appears to be wrong from beginning to end. According to Mark Simonsen at Beagle Bros, Claris made some last-second changes to AppleWorks 2.1 that prevented the third-party updates the companies had prepared for AppleWorks 2.1 from working. At the moment, it appears you need Applied Engineering's **AW 2 Expander** v3.0.1; Beagle Bros **Timeout** v2.1; Jem Software's v2.1; and you have to stick with AppleWorks 2.0 if you want to use **Pinpoint**. Update at 11 (as in November).

On last month's page 4.64, I mentioned the programs **Iconix** and **Sonix** from So What Software, but neglected to include prices or a phone number. **Iconix** is \$39.95, **Sonix** is \$49.95, and So What's phone number is 714-964-4298.

Back in July, (page 4.45) I neglected a phone number for Innovative Systems, the company that's producing the floating point card for Apple IIs that uses the 68881 math chip. You can reach Innovative at 301-987-8688 or 301-768-4599.

Apple's September dealer service notes have more complete information about the problems Apple had with RAM chips last New Year's Eve. I mentioned this in August (page 4.49). The problem concerns both IIGs memory expansion cards and IIGs memory expansion kits. The problem is that some of the chips Apple used weren't "CAS before RAS" (so even Apple makes this mistake!). There were two types of bad chips. One type has a "NEC" manufacturer code and, just below it, a "UK" country of manufacture code. The other set has a "NEC" manufacturer code, a "JAPAN" country code just beneath it, and a four-digit date code to the right of "JAPAN" followed by a "P" and some more stuff. If you have a memory expansion card with chips like these soldered onto it, your Apple dealer will replace it free. If you bought an Apple memory expansion kit with chips like these (chips bought from Apple have an "A" marked in the lower left corner of the chip), your Apple dealer will replace them.

(Apple's September's service notes also said that beginning this month the Apple IIGs will be shipped with a new operating system called GS/OS on system disk 4.0. Older IIGs units will require the IIGs ROM upgrade (see September 1987, page 3.57) to use this software. Shhhh...we're not supposed to know this.)

Also in August (page 4.50) I left a comma out of line 50 of the program that checks for CyberAIDS. It goes between the open-quote and the "A\$". And on page 4.54, third column, second paragraph I gave two of the Manx C

compilers the same name. Aztec C65-d is the correct name for the DOS 3.3 system; the ProDOS system is called Aztec C65-c.

Way back in March (page 4.15), I dropped an "S" from the answer to the letter "Overstrike cursor at startup". It goes between "APLWORK" and ".SYSTEM".

A representative of Orange Micro was our guest at a real time conference on GEnie on August 30; we found out the answer to the question asked in last month's letter, "Orange Micro's Support" (page 4.58), is to get a v1.1 ROM for the ImageBuffer from Orange Micro.

And, finally, TIM.SWIHART, one of the leaders of the Apple II Programmers and Developers RoundTable on GEnie, sent me an electronic pie embedded with a message that said my answer to last month's letter "UniDisk Tech" (page 4.59-60) was incomplete. Full information on UniDisk 3.5 internals is also available in the **Apple IIGs Firmware Reference Manual**, pages 142-151.

Back to bent arrows

I read in the September issue that you are in search of a way to get the AppleWorks word processor to display the MouseText bent-arrow character for Return instead of the fuzzy box. I've uploaded a file called AW.LOCATION.BNY to GEnie that contains this patch, as well as many others. Or you can use another of my uploads, SUPERPATCH, to install these patches into AppleWorks 2.0.

John Link
Kalamazoo, Mich.

Here's the patch. You have to change three locations:

```
POKE 768,205
BSAVE SEG.M1,T$00,A768,L1,B$6FE1
BSAVE SEG.M1,T$00,A768,L1,B$7175
BSAVE SEG.M1,T$00,A768,L1,B$7A91
```

Link's files are probably available on other online services by now as well.

Another bent arrow

Another good place to use the MouseText bent arrow in AppleWorks is in place of underline cursor. It makes it much easier to see. To make this change do this:

```
POKE 768,77
BSAVE APLWORKS.SYSTEM,TSYS,A768,L1,B$DA1
```

Thomas Militello
Rancho Palos Verdes, Calif.

We've given essentially this same information before (see "Blink and its gone", May 1987, page 3.32 and "Changing the blink speed", September 1987, page 3.63). The important new information your tip ads is that you can replace the insert cursor with any of the MouseText characters by using values in the 64 to 95 range.

Update blues

I think I am going to write a book and entitle it "Adventures in Upgrading". Getting AppleWorks 2.1 to work with my other software has taken a lot of postage and aggravation. At the rate I am going with problems with JEM, Beagle, and Applied Engineering, it will probably be Christmas or maybe even AppleWorks-GS before I get v2.1 installed. The only thing so far that has been easy about upgrading has been the response from Claris. I could not believe how quickly they provided the update. Their

parent never responded so well.

Whit Crowley
Manchester, Mo.

Upgrades are always a hassle, especially for those who are near the front of the line. I've been known to skip a revision level every once in awhile, but I'm not convinced even that really helps.

Don't pass go, v2.1

I thought you might like to have the patches to bypass the startup "press-space-bar-to-continue" and "enter-current-date" messages for the new AppleWorks 2.1:

```
BLOAD APLWORKS.SYSTEM,TSYS,A$2000
POKE 14118,44 : REM bypass "press space bar"
POKE 14436,208 : REM bypass "enter date"
POKE 14437,19
BSAVE APLWORKS.SYSTEM,TSYS,A$2000
```

Daniel R. Creech
Hannibal, Mo.

Desktop expansion

What are three economical ways of expanding AppleWorks desktop memory?

T. Garner
Coquitlam, B.C.

There are only three ways. With the price of RAM chips, none of them seem very economical. They are "aux-slot", "standard-slot", and "memory-slot" RAM cards.

Ile owners can buy either a standard-slot or an "aux-slot" memory card. Examples of **standard-slot** cards are Applied Engineering's RamFactor, Apple's Memory Expansion Card, and Cirtech's PR cards. AppleWorks 1.3 and higher automatically recognize this type of card and use it for desktop expansion. Examples of **aux-slot** cards are Applied Engineering's RamWorks and Checkmate's MultiRam. AppleWorks must be patched, using software provided with these cards, to recognize aux-slot memory. The patch programs, however, provide word processor and database expansions beyond what AppleWorks itself provides for standard-slot cards.

IIC owners have the same options as Ile owners. However, the only standard-slot-type card available for the IIC is Apple's own. Older IIC models require a free motherboard upgrade to use this card. Applied Engineering and Checkmate Technology both make aux-slot-type cards for the IIC. AE's is called Z-RAM and Checkmate's is MultiRam CX.

IIGs owners will find that AppleWorks 2.0 and higher automatically expand into their **memory-slot** cards. These are made by a number of companies. A IIGs can also be fitted with a standard-slot-type (but not an aux-slot-type) memory card, but AppleWorks will not expand into it without a special patch (see "AppleWorks IIGs defeater", December 1987, page 3.86 and "IIGs defeater defeated", January 1988, page 3.95.)

Device list too short

I would like to have the AppleWorks "Disk drives you can use" list to display the choice "Disk 1 (Slot 2)" for my second 3.5 drive. To date neither I nor my dealer nor the Washington Apple Pi hot line nor Claris technical support

can get it to do this with my hardware configuration. It doesn't matter whether I use plain vanilla AppleWorks or a patched AppleWorks. It also doesn't matter if I start Appleworks from my hard disk or 3.5 drive.

In addition to my 20 meg Sider in slot 7 and two 3.5s, I have one 5.25 in slot 6 and a 1.5 meg Iigs internal RAMdisk that shows up as slot 5, drive 2.

Charles O. Ward
Centreville, Va.

Your problem is that AppleWorks will display only six devices in its "Disk drives you can use" list. You have more than that. The six that show up in your list are the two volumes in slot 7, two in slot 6 (ProDOS can't tell if you have one or two 5.25 drives connected, so it always assumes two), and two in slot 5 (3.5 and /RAM5). Your seventh, the 3.5 that appears to be connected to slot 2, won't fit in the list.

One solution is to access the second 3.5 by name using the "ProDOS directory" choice at the bottom of the list.

Another solution is to tell ProDOS you really only have one 5.25 drive connected. If you have Glen Bredon's **ProSEL**, his SCAVENGE program will remove the phantom drive from the ProDOS device table. The newest version of **ProSEL's** CAT.DOCTOR also has this ability from its auxiliary (CD.EXT) menu (\$40, 521 State Rd, Princeton, NJ, 08540).

If you know a little assembly language, another possibility is to get a copy of Apple's **ProDOS Technical Note #8** (last revised 12/85) from APDA or an online service. It explains how to remove drives from the ProDOS device list.

<L>ayout, search macro

Do you think most AppleWorks users know about using open-apple-<Layout to modify the single-record screen in the database? I've been using AppleWorks for almost three years and just stumbled upon it (it gets a mere half page in the old manual). At last I have a readable format for bibliographic entries.

The most useful word-processing macro I've thought of is:

S:<oa-1><sa-F><key><key><key><key><key><run><rtn>!
note: <sa-F> is <oa-F>t<oa-Y>

Press solid-apple-<S>earch and five characters and the macro will take you to the first occurrence of those five characters in the document. I find it useful when I revise long documents on paper and then enter the changes in AppleWorks. You could make the string in the macro as long as you want; I've found five characters virtually always enough. You can do a lot to make sure that they are enough by typing a string that contains the end of one word and the beginning of the next. The final <rt> is neat since it works for both <lt> and <gt;, I don't want to find the next occurrence, and <Space Bar>, as in 'Not Found, press Space Bar to continue'.

Michael Leddy
Charleston, Ill.

Since almost all of the public-domain templates I've seen have nothing more than the stock two-column layout for the single-record screen, I, like you, suspect most people don't know they have the capability of rearranging the positions of the categories on that screen.

Some of the AppleWorks enhancement programs that add mouse support to AppleWorks allow you to use the mouse to layout the cate-

gories on the screen. For my time, this is the best use of a mouse inside AppleWorks.

Exclaimed vertical lines, \$300

In May a Swedish Apple II user asked how to change the vertical lines in AppleWorks so they used the exclamation point instead of ASCII 124, which isn't a vertical line in the Swedish character set but an o with two dots over it. Here are the patches for the menu cards and the oa-Q menu. I wasn't able to find the needed locations for the Database/Spreadsheet vertical lines or for Tab Stops. (Try looking for ASCII 124 in AppleWorks and you'll find it appears hundreds of times.):

```
POKE 768,33 : REM "!" char
BSAVE SEG.M1,TS00,A$300,L1,B$14B0F
BSAVE SEG.M1,TS00,A$300,L1,B$14B6F
BSAVE SEG.M1,TS00,A$300,L1,B$14BB0
BSAVE SEG.M1,TS00,A$300,L1,B$14C02
BSAVE SEG.M1,TS00,A$300,L1,B$14C05
BSAVE SEG.M1,TS00,A$300,L1,B$15836
BSAVE SEG.M1,TS00,A$300,L1,B$15842
```

As for the space at \$300, both *Timeout* and *UltraMacros* use parts of that page. However, \$300-\$31F is still safe from both these two additions.

Mark Munz
Fort Lewis, Wash.

The ASCII 124 problem was encountered in Greece two years ago when we were localizing AppleWorks. About 12 other ASCII codes also caused problems. I used ProSEL's BLOCK.WARDEN and recommend the following procedure:

- a.** Make lots of AppleWorks disks.
- b.** Prepare yourself to stay sleepless for a week or so.
- c.** Scan SEG.M0 and SEG.M1 for the assembly commands LDA, LDX, and LDY followed by ASCII 124 (that's \$A9 7C, \$A2 7C, and \$A0 7C). Change them one-by-one and try AppleWorks every time. Not all are meant to be changed. It took me a full month's work to change all of AppleWorks into Greek.

Nick Andritsakis
Infostar Computer Consultants
Kallithea, Greece

The lesson here for software authors is to set up a table in your program that holds all the changeable characters. Instead of doing an "immediate" register load, get the special characters from the table. Localizers could simply scan for and change the table. The characters that change from language to language are ASCII 35 (#), 64 (@), 91 ([), 92 (\), 93 (]), 96 (`), 123 ({), 124 (|), 125 (}), 126 (~), and 127.

Czech video ROM needed

I need to be able to operate a database in the Czech language. My Ile has a German keyboard and a switch that lets me flip to an American keyboard. Where can I get a chip that would let me switch between German and Czech? I also need to find a supplier of European alphabet daisy wheels for my printer. Any ideas?

Myron E. Schirer
Vienna, Austria

According to Jim Sather's **Understanding the Apple IIe** (now out of print), the video ROM in your Apple IIe is an 8K, 28-pin 2764 ROM (IIes sold in the U.S. use a smaller, 4K video ROM). We know that it's technically possible to replace it with an EPROM that holds the German and Czech character sets, but we don't know where you'd get one already made

up. We don't know of a source for the daisy wheels you need, either, but will be interested in what our European subscribers know about this.

Old DOS, new CAT

Finally, the CAT command comes to DOS 3.3:

CALL -151
9D4C:6D N A8F1:43 41 N A937:40 70
3D0G

This changes the INT command to CAT. CAT does the same thing as CATALOG. CATALOG still works, too. INT doesn't.

Mark Cornick
Charlottesville, Va.

RESUME doesn't

A DOS 3.3 quirk I've run into and never seen an explanation of is the use of RESUME after an ONERR GOTO when there is an I/O ERROR after a text file has been partially read. RESUME doesn't work—it just prints a "?" on the screen. If there are still N records in the text file to be read, you can hit Return N times and then the file is closed and the program continues. Any ideas?

John Waters
Tampa, Fla.

Under DOS 3.3, an I/O ERROR will turn off your READ and close your file. When you RESUME, you go back to the INPUT statement that was being executed when the I/O ERROR occurred. Since READ is no longer active, INPUT looks to the keyboard, rather than the file, and a question mark appears on your screen.

Most programs simply declare the file unreadable at this point. If you want to be more sophisticated, your error handling routine would have to reOPEN the file and issue another READ command. This, of course, would set the file pointer back to the beginning of the file, not to the spot you were reading when the error occurred. You could track how many characters have been received as a file is read and use a B parameter with your READ statement to go back to the exact byte that caused the I/O ERROR. See July 1985, page 1.51, for more on the B parameter. Also see January 1985, page 1.02, for more on ONERR GOTO with READ.

Assembly RUN

I know that you can RUN an Applesoft program from assembly language by jumping to \$D566 (54630). This will execute any Applesoft program in memory from the beginning, as if the RUN command was entered. I'd like to be able to execute a program from any line—is there a way to RUN a program starting from a specific line number? Perhaps a GOTO would be better, as the variables could be retained.

Benjamin Ng
Calgary, Alb.

Make sure that Applesoft has been initialized (see August, page 4.56) and that your program has been loaded, then:

```

A9 YY      LDA #YY      Store $XXYY in $50-51
85 50      STA $50
A9 XX      LDA #XX
85 51      STA $51
4C 55 D9   JMP $D955

```

This is the equivalent of a GOTO \$XXYY command, where \$XXYY is a line number (in hex).

ProDOS zero-page usage

I'm writing a program on my IIe using assembly language under ProDOS. Where can I find a listing of zero page locations used by ProDOS so that I don't accidentally clobber the area my program is using with ProDOS (or vice-versa).

Allan G. Dunn

The ProDOS Machine Language Interface uses locations \$40 through \$4E, but it restores them to their original values before a call is completed. This is why you never see much about ProDOS zero-page usage. The floppy disk driver routines inside ProDOS also use \$3A through \$3F. These are not restored and should be avoided if your program will support 5.25 inch disks.

Directory deletion

When I try to delete a subdirectory from a ProDOS disk from Applesoft it says FILE LOCKED. Even if I unlock it I have the same problem.

Chang Yuh Kang
Singapore

FILE LOCKED doesn't describe the actual problem—it was just an error message already built into Basic.system. The real problem is that you can't delete a subdirectory from Applesoft unless it's empty. First delete all the files in the subdirectory, then delete the subdirectory. The FILE LOCKED error will disappear along with your subdirectory.

Undelete problems

Yesterday, while doing some house cleaning of disks to remove old files, I inadvertently deleted a needed file. I tried to undelete using Copy II Plus, but couldn't. In what way does AppleWorks delete a file so that it cannot be recovered?

Donald Bock
Hudson, Fla.

It depends which version of ProDOS you were using with AppleWorks when you deleted the file. Files deleted while using versions of ProDOS prior to 1.3 cannot be recovered. How's that for a good reason to update your disks with a newer version ProDOS?

ImageWriters in the office

Today we had a label stuck underneath the platen of one of our ImageWriter IIs. We spent more than an hour with screwdrivers, tweezers, pliers, a pencil compass, an Exacto-knife, and anything else we could think of trying to get that little sucker out of there (we have a blowtorch downstairs—I was tempted, but managed to resist).

I even spent about half an hour trying to figure out how to just remove the bloody platen from the printer, but nothing about the task is obvious. While I sat there with sweat pouring off my face, one of the secretaries brought me a bottle of something called Dr. Scat! typewriter cleaner. I was very skeptical, especially because the applicator ball was far too big to squeeze under the platen. But I tried it, squeezing some of the fluid into the crack. Then I rolled a piece of paper through the printer and said "WOW" as an inch-square piece of label rolled right out with the paper.

I don't even know where to get the stuff, but

it's probably available in office supply stores. I am thoroughly impressed with it. I gave the whole platen and print head a good cleaning, too, and the stuff did a marvelous job (it's unbelievable how filthy a year-old printer can be). I really don't know if this brand of cleaner is the best, but I certainly know that it works.

Here's a handy AppleWorks tip I've never seen in print before. In our office we do a lot of stuff on single-sheet paper, though we don't have a cut sheet feeder. It's easy enough to roll the continuous paper back and flip the single-sheet latch—the problem was that AppleWorks was configured for continuous paper and would expect the next sheet to be there when it wasn't. First I tried to teach our secretaries to use AppleWorks's open-apple-<O>ption Pause Each Page feature to solve this, but they often forgot to include that code when printing on single sheets.

So I configured AppleWorks for two printers. The specifications for the two printers are identical except that one stops at the end of each page. I named them "ImageWriter II" and "Image Pause". Our secretaries have a much easier time remembering to pick the proper printer than they did remembering to use Pause Each Page.

Dean Esmay
Flossmoor, Ill.

You could probably even name the printers "continuous paper" and "single sheets". Those make sensible answers to the question "Where do you want to print the file?"

You hint at another useful tip here, but don't actually say it—when switching between continuous paper and single sheets on an ImageWriter II, you don't have to completely remove the continuous paper from the printer. When you flip the single-sheet switch, it disconnects the pinfeed tractors, so you can leave continuous paper "threaded" as long as it's clear of the platen area.

Print downhill no more

I read with interest last month's letter called "Printing downhill" (page 4.59). My Epson MX-80 used to print with a downward drift in double-strike mode when each character was being underlined separately. A magnifying glass revealed the reason: Prior to the second pass for printing each underlined character, the paper would advance a fraction of a dot and the second strike would fill in white space between vertical dots. A fraction of a dot is not much, but when it happens for very many characters, it is decidedly noticeable.

The answer is to use the Epson's underlining feature, not Apple Writer's. The paper still advances for overstrike but only once for the entire line rather than once for each character underlined.

Robert H. Holdsworth
Wilbraham, Mass.

To use the Epson's underlining feature from within Apple Writer, enter control-V, the Epson's underline-on code, and another control-V. Use the same sequence to turn it off. The control-V allows you to enter control codes into your text. Back when I used Apple Writer all the time I had some "glossary keys" or macros set up to enter all the codes; all I had to remember was which macro turned underline on and which turned it off.

To avoid having underlines sticking out past the left margin when a line breaks while underline is on, you also need to use the printer's

left margin command and leave Apple Writer's left margin set to zero.

I looked in our Programmers' Handbook of Computer Printer Commands for the Epson underline on and off codes. It doesn't show any for the MX-80, which could be a significant limitation of this technique for some Epson owners. For other Epson printers, including the Epson FX that started all this, underline on is "ESC-I" (the hyphen is part of the command) and underline off is "ESC-O". The left margin command is "ESC L n", where "n" is the width of the left margin in characters. (The width in inches depends on what character set you are using.)

Mini-8 ImageWriter dealers

A cable is almost always a cable when referring to the Mini-DIN-8 (IIgs modem cables, cont', September 1988, page 4.58). But if you use a switchbox to run more than one printer/computer combination you will get into trouble if you use all standard mini-8 cables. Whether you have one computer and several printers or one printer and several computers, the cable from the box to the single device cannot be a normal mini-8, it must be a special straight-through cable. If not, the combination will not work.

ImageWriter IIs can be very crabby about paper. The symptoms described in "Chronic printer problems" (page 4.59) and earlier letters can be a result of paper that is too thin or too thick. Too thin paper doesn't push well, especially when the humidity is high. Extra thick paper also can be too hard to push through the printer. It's also possible that the paper is hitting the bail and binding when the printing begins.

ImageWriter II paper drive motors can fail and still work sorta, kinda. This failure is characterized by a horrible grinding noise when the paper should be advanced, but will not be consistent. Sometimes the paper will feed, sometimes not. The fix is to replace the motor.

Incidentally, I'm an Apple dealer and I disagree with a statement in the letter "Repair Restraint" (page 4.62). Apple does not specifically require a dealer to use Apple parts for the repair of Apple computers. Most of the parts simply aren't available anywhere else. The prices aren't cheap, but we've found Apple to be a fast and fair supplier of service parts. I wish I could say that about most of the electronics companies we deal with.

Power supplies and TTL logic chips are the most notable items that are available from sources other than Apple. We have tested and sold non-Apple power supplies for the II-Plus and IIe. Our experience is that yes, they are cheaper, but they also have a shorter life span. We have also had a defect rate approaching 25 per cent.

Very little component-level service is done anymore on logic boards, so we consume few TTL ICs these days. Apple doesn't care where we obtain these parts.

Vern Mastel
Bismarck, N.D.

I added an automatic printer switch to our collection of equipment this month and the cable connections made me crazy. The switch connects four computers to one printer. The computers are a IIe, IIc, IIgs, and Mac SE, so there's quite a combination of connectors. The

printer's an ImageWriter II. The printer switch I used had standard RS-232 25-pin connectors.

It took forever to wire up all the cables. As you predict, my carefully thought-out plans for the cables, using wiring diagrams from a number of books, wouldn't work. I soldered and unsoldered cables until the wee hours of the morning before I got things going.

The printer switch I used automatically detects when any of the computers starts to print something and connects it to the printer. While that computer has the line, the printer appears 'not selected' to the other three computers, so if you try to use one of them to print something it will politely 'hang' until the first computer is finished. Another feature of the box is that it has lights that flash constantly, which is much more impressive to friends and neighbors than any other equipment I have.

The company I bought the switch from said it wouldn't work with an Apple II (they wanted me to buy a twice-as-expensive unit with a memory buffer), so I'm not going to recommend them. If you know of anyone who produces a switch like this with mini-8 connectors and decent wiring diagrams, I'd like to know about it.

Printing double-wide

What are the control codes to get an ImageWriter II to print 4 and 6 characters per inch?

Clen Clogan
Monticello, Ill

To get characters this big you have to use the ImageWriter's codes for double-wide characters. You can't actually get 4 characters-per-inch. But you can get 4.5 by setting the ImageWriter for 9 cpi and double-wide. 6 cpi is the ImageWriter's 12 cpi and double-wide.

For example:

```
10 DS=CHR$(4) : ESC$=CHR$(27)
20 PRINT DS;"PR#1" : REM printer on
30 PRINT ES;"n"; : REM 9 cpi
40 PRINT CHR$(14); : REM double-wide
50 PRINT "This is 4.5 chars/inch."

60 PRINT ES;"E"; : REM 12 cpi (elite)
70 PRINT "This is 6 chars/inch."

80 PRINT CHR$(15); : REM double-wide off
90 PRINT "This is 12 chars/inch."
95 PRINT DS;"PR#0"
```

SCSI numbers don't add

In the July issue on page 4.46 you say that up to seven SCSI devices may be connected to one SCSI card. I purchased a CT-20 hard drive from Chinook Technology, but the User Manual indicated only two devices could be connected. Which is correct?

Gary Mertl
Brookfield, Wisc.

SCSI allows for seven devices per SCSI chain. Current versions of ProDOS, however, only allow two devices per slot. This is where the confusion comes from. I expect future versions of ProDOS will leap over the two-devices-per-slot limitation.

Another view on cheap drives

In your reply to "Rethinking hard disks, (cont.," in the August issue (page 4.53) you say IBM-type drives aren't really cheaper in the long run and that all the letters you've received about the Perlin Megaboard have been negative.

I have found neither to be true.

I purchased a Megaboard about a year ago and have been very satisfied. I am using a Tandem TM-703 drive that I obtained surplus for \$200. It is an AT-class drive that formats (IBM) to 32 megabytes, has an average access time of 45 milliseconds (fast), and has auto-park. With a case and power supply available for as little as \$50 (I paid \$75 for extra current capacity) and with the Megaboard (which includes all cables and software) for \$195.00, I was online for under \$475.00.

The only drawback to the Megaboard is that its hardware is set up for drives with 4, 6, or 8 heads. As about 80 per cent of the drives available fall into one of these categories, it's usually not a problem. The TM-703, however, has 5 heads. This means that I can only access about 25 megabytes on my drive. Due to its speed and other features, I find that limitation acceptable.

The Megaboard allows partitions for DOS 3.3, ProDOS, CP/M, and Pascal. I find the system quite fast and flexible. If I had it to do over again, I would still have purchased the Megaboard, although a drive with an even number of heads would have been better.

I feel I must correct your other impression, as well. In addition to several Apples, I also own an IBM clone. I added a 32 megabyte hard drive to it also. The drive was purchased for \$325 and included a controller card and cables. That's all you need. No upgrades to the power supply or operating system were needed, contrary to your statement. IBM power supplies include connectors for four drives, hard or floppy. MS-DOS (2.0 or higher), which is included in most system purchases, is fully capable of accessing hard drives without modification. My IBM drive is an admittedly slow 65 millisecond model, but it has worked fine for about two years.

I find each machine has its own set of advantages and disadvantages and that one of each is the best way to go.

John L. Alexander
PFO San Francisco

I reviewed the major complaint letter we've received about the Megaboard and have to admit it contains a lot more smoke than fire. In addition to requiring an even number of heads, the other big complaint is that on a IIgs the Megaboard has to go into slot 6, which makes it difficult to move files from 5.25 floppies to the hard disk. There are also some limitations to the DOS 3.3 partition, which has a maximum size of 38 140K volumes per drive. 400K DOS 3.3 volumes are not supported. There was also a problem with the configuration software, which couldn't deal with drives larger than about 43 meg, but that bug has been fixed. The software now works with drives up to 64 megs in size.

As for the IBM drives, we priced all the various items you need separately. If you are able to buy them in a bundle or get them when you buy your system, as you did, the prices get better. We've always agreed that IBM-type drives are cheaper than Apple drives; our point is simply that the real price difference is less than the perceived difference—users who put together a hard drive system using IBM-type drives **will** save money, but usually not as much as it would at first appear.

French accents

Using an Apple IIgs and ImageWriter II, I have been unable to access French vowels with circumflexes (the vowel with a caret above it). All other accented vowels are available in the French character set on the ImageWriter, and via the control panel on the IIgs, but not with any software I've yet tried (AppleWorks, Word Perfect GS, MultiScribe). I know it's accessible in Word Perfect for IBM. How do Apple users manage?

Dorothy Nesbitt
Winnetka, Ill.

Let's start with the AppleWorks example. Go into the 'Options' selection in the IIgs Control Panel and set the screen display to French. You may also be interested in setting the keyboard layout to French, which is a separate selection. Now AppleWorks will display the built-in French characters, which, as you point out, do not include circumflexed vowels.

To get this much to print on your printer, you have to tell the ImageWriter you want to print in French. You can do that 'permanently' by setting the dip switches correctly, or with an AppleWorks printer setup (see "Danish to go" in our July issue, page 4.47, for more information).

In France, I suspect the ImageWriter adds the circumflex by backspacing over the vowel and printing it separately. There is no way to duplicate this on the IIgs text screen and it is a feature that the U.S. version of AppleWorks doesn't support. I'll rely on our French subscribers to fill us in on how this works with the French version of AppleWorks.

With a program like **MultiScribe** you are not limited to the IIgs text screen or the ImageWriter's built-in fonts. What you need is simply a font that contains all the French characters. I'm not sure whether there's a French font in the public domain, but if there is I'm sure we have it available for downloading on GENie. There may be problems with display of even special fonts, however, as described in the next two letters.

To summarize, I don't know very much about this, but I know we have subscribers in Europe who are very familiar with the issues and who'll tell us how to proceed.

More characters, more tongues

HodgePodge, from the Programmer's Introduction to the Apple IIgs disk, let's me see and print all characters of any font, from \$00 to \$FF, including accented letters, symbols, and dingbats. But all other ProDOS 16 applications let me see and print only from \$20 to \$7E.

Is it possible to use all characters with MultiScribe, Draw Plus, Graphic Writer, or PaintWorks? And why don't these applications obey the 'Display Language' and 'Keyboard Layout' options of the IIgs Control Panel?

I also tried to read, with MultiScribe 3.0, an ASCII file with accented characters taken from a Mac, but MultiScribe resolutely refused.

Luigi Bruno
Rome, Italy

There is one aspect of some IIgs software that I have not seen mentioned in any article I have read. In Macintosh software it is possible to generate extra characters by holding down the option key when typing. This is not possible in all IIgs software—for example, it doesn't work in MultiScribe GS (at least up to 3.01c), but it does in TopDraw. These two programs are from

the same company!

Mac and GS fonts use 8-bit ASCII. Normal characters are "low ASCII" and the extra characters that are available correspond to "high ASCII". These extra characters are most easily seen using *HodgePodge*.

I asked *StyleWare* about this nearly a year ago and their response was "the lack of ability to access the high ASCII set is a limitation of the operating system of the Apple IIgs. In *TopDraw* it was possible to overcome this limitation without corrupting the system files, but we have yet to find a way to do this in *MultiScribe*." I also tried asking Apple Australia about this but did not get anywhere.

It's possible to overcome this problem, but it's tedious. First you need to create a file that has the 128 low-ASCII characters in it. Do this with *MultiScribe* or whatever. Next, use a zap program to change each character to it's high-ASCII equivalent. Once you have a file like this (you could also use a program that will deal with high-ASCII characters, like *TopDraw*, to create it) you can use copy and paste via the clipboard to put the special characters into your document.

On another subject, in reference to "Speaking in tongues" in your August issue, you didn't mention *Pecan's Power System* software. I haven't used this software or seen any reviews, but I have the following information from their brochures. UCSD Pascal, Modula-2, FORTRAN-77, and BASIC compilers are available, as well as a separate assembler. Each of the five is available in either an "Apple II 6502" version or a "Apple IIgs" 65816 version. All of the compilers use SANE numerics and can "access ProDOS files", although the compilers don't appear to be

ProDOS-based. The brochure says the IIgs implementations utilize extended memory, support toolbox routines, and are compatible with Apple Pascal. A variety of utilities are available. I was interested because I would like a FORTRAN-77, but am still undecided. I would prefer a compiler that ran under APW.

Stephen Harker
Oakleigh, Vic.

If *HodgePodge* can display all 256 characters, it would seem other programs should be able to. We'll see what we can find out.

Four compilers and an assembler times two versions (Apple II and Apple IIgs) makes ten separate packages from *Pecan Software*. Each package sells for \$100. We haven't seen any of these products either. (*Pecan Software Systems*, 1410 39th St, Brooklyn, N.Y. 11218 718-851-3100 or 800-637-3226.)



Watch those controls

After we upgraded the ROMs on my father's IIgs, trying to turn on the 80-column text screen from Applesoft with PRINT CHR\$(4); 'PR#3' yielded a NO DEVICE CONNECTED error. PR#3 in immediate mode did the same. It turned out to be a control panel problem—slot 3 was configured for Your Card.

Clark Hugh Stiles
Grand Rapids, Mich.

CDA/NDA lesson

I have downloaded some Classic Desk Accessories from GENie, but I haven't seen how to install them or how to access them. Could you fill in the details?

John Christensen
Indian Head, Md.

Desk accessories come in two flavors "classic", or CDA, and "new" or NDA. They are useful only on the IIgs; earlier Apples can't use them.

Take your CDA and NDA files and copy them to your ProDOS 16 System Disk (the disk you boot from). They **must** go in the SYSTEM/DESK.ACCS folder (subdirectory).

When you next boot that system disk, the desk accessories will be automatically installed.

To use the classic ones, press open-apple/control/escape, just like you were going to use the control panel. You will see a list of the CDAs and you can choose which ones you want to run. You can access this menu from any kind of software, including newer ProDOS 16, older ProDOS 8, and DOS 3.3 packages (unless the program turns off "interrupts", in which case the menu isn't accessible).

To use new desk accessories, you must be

using a ProDOS 16 program, such as *Finder*. Point at the small apple in the upper-left corner of the screen and press on the mouse button. A menu will drop down that includes the NDAs on the disk you booted from.

There are limits to how many desk accessories you can have access to at one time. However, those limits have been punctured by programs such as **Two Apples** and **Master CDA**, mentioned here in July ("Desk accessory limits, page 4.45).

Beagle Compiler and slot 2

Since the time I wrote the "IIgs is slot poor" letter that you published in the September issue (page 4.62), Alan Bird has modified the Beagle Compiler such that APPEMEM.SYSTEM now recognizes a memory card in slot 2.

Elliot Lifson
Yonkers, N.Y.

The bias potential

I request that my subscription be cancelled immediately and that you refund the prepayment amount on my subscription. While I don't owe you a reason, I will volunteer one. While I cannot be knowledgeable as you or your staff about the bottom line in what it costs to run such a newsletter as yours, I have become painfully aware in recent months of the shift in emphasis on sideline retailing of books. When I received this month's issue and saw the attention given to the pitching of memory cards, I decided it was time to part company.

I support *Consumer Reports Magazine* for the same reason I've supported you in the past. I believe there is a place for newsletters and magazines of small, select audiences that do not accept advertising and do not sell the products they are talking about. You've crossed the line. And lost a subscriber.

Kate Kelbaugh
Herndon, Va.

I don't think it's fair to compare **Open-Apple** to **Consumer Reports**. **Consumer Reports** is a mass-market publication that is ten to twenty times larger than our newsletter, online services, and book and product sales combined. It solicits, receives, and depends on tax-deductible donations for its existence.

But I'm sorry if I created the impression somewhere along the line that **Open-Apple** would be designed on the **Consumer Reports** model. **Consumer Reports** is not just a different animal from what we're trying to do here, it's a different species.

Open-Apple is a "high-tech/high-touch" Information Age enterprise. What we do for a living, at its most essential level, is bring people with a common interest together so they can help each other accomplish their goals. This is easy to see even in the Cirtech case, where we're giving U.S. Apple II users the opportunity to obtain high-quality products made by European Apple II users.

On the other hand, I recognize that selling both products and information about products creates an explosive mixture. I'm acutely aware that we have to handle it carefully. If we ever start to let our **newsletter** (rather than our catalog) recommend what we sell, rather than what's worth recommending, we'll dilute the value of our information. I don't want or intend to do that and I appreciate people like you making every effort to keep me honest.

Open-Apple

is written, edited, published, and

© Copyright 1988 by
Tom Weishaar

with help from

Tom Vanderpool Sally Dwyer
Dennis Doms Steve Kelly

Most rights reserved. All programs published in **Open-Apple** are public domain and may be copied and distributed without charge. Apple user groups and significant others may obtain permission to reprint articles from time to time by specific written request.

Open-Apple has been published monthly since January 1985. World-wide prices (in U.S. dollars; airmail delivery included at no additional charge): \$28 for 1 year; \$54 for 2 years; \$78 for 3 years. All back issues are currently available for \$2 each; bound, indexed editions of our first three volumes are \$14.95 each. Volumes end with the January issue; an index for the prior volume is included with the February issue.

Please send all correspondence to:

Open-Apple
P.O. Box 11250
Overland Park, Kansas 66207 U.S.A.

Open-Apple is available on disk for speech synthesizer users from Speech Enterprises, P.O. Box 7986, Houston, Texas 77270.

Open-Apple is sold in an unprotected format for your convenience. You are encouraged to make back-up archival copies or easy-to-read enlarged copies for your own use without charge. You may also copy **Open-Apple** for distribution to others. The distribution fee is 15 cents per page per copy distributed.

WARRANTY AND LIMITATION OF LIABILITY. I warrant that most of the information in **Open-Apple** is useful and correct, although drift and mistakes are included from time to time, usually unintentionally. Unsatisfied subscribers may cancel their subscription at any time and receive a full refund of their last subscription payment. The unfilled portion of any paid subscription will be refunded even to satisfied subscribers upon request. MY LIABILITY FOR ERRORS AND OMISSIONS IS LIMITED TO THIS PUBLICATION'S PURCHASE PRICE. In no case shall I or my contributors be liable for any incidental or consequential damages, nor for ANY damages in excess of the fees paid by a subscriber.

ISSN 0885-4017
Printed in the U.S.A.

GENIE mail: OPEN-APPLE
913-469-6502